Internet Archive Blogs
A blog from the team at archive.org

Downloading in bulk using wget

Posted on April 26, 2012 by internetarchive

If you've ever wanted to download files from many different archive.org items in an automated way, here is one method to do it.

Here's an overview of what we'll do:

1. Confirm or install a terminal emulator and wget
2. Create a list of archive.org item identifiers
3. Craft a wget command to download files from those identifiers
4. Run the wget command.

Requirements

Required: a terminal emulator and wget installed on your computer. Below are instructions to determine if you

already have these.
Recommended but not required: understanding of basic unix commands and archive.org items structure and terminology.

Section 1. Determine if you have a terminal emulator and wget. If not, they need to be installed (they're free)

1. Check to see if you already have wget installed
If you already have a terminal emulator such as Terminal (Mac) or Cygwin (Windows) you can check if you have wget also installed. If you do not have them both installed go to Section 2. Here's how to check to see if you have wget using your terminal emulator:

1. Open Terminal (Mac) or Cygwin (Windows)
2. Type "which wget" after the $ sign
3. If you have wget the result should show what directory it's in such as /usr/bin/wget. If you don't have it there will be no results.

2. To install a terminal emulator and/or wget:
Windows: To install a terminal emulator along with wget please read Installing Cygwin Tutorial. Be sure to choose the wget module option when prompted.

MacOSX: MacOSX comes with Terminal installed. You should find it in the Utilities folder (Applications >

Utilities > Terminal). For wget, there are no official binaries of wget available for Mac OS X. Instead, you must either build wget from source code or download an unofficial binary created elsewhere. The following links may be helpful for getting a working copy of wget on Mac OSX.
Prebuilt binary for Mac OSX Lion and Snow Leopard
wget for Mac OSX leopard

Building from source for MacOSX: Skip this step if you are able to install from the above links.
To build from source, you must first Install Xcode. Once Xcode is installed there are many tutorials online to guide you through building wget from source. Such as, How to install wget on your Mac.

Section 2. Now you can use wget to download lots of files

The method for using wget to download files is:

1. Generate a list of archive.org item identifiers (the tail end of the url for an archive.org item page) from which you wish to grab files.
2. Create a folder (a directory) to hold the downloaded files
3. Construct your wget command to retrieve the desired files
4. Run the command and wait for it to finish

Step 1: Create a folder (directory) for your downloaded files

1. Create a folder named “Files” on your computer Desktop. This is where the downloaded where files will go.

Create it the usual way by using either command-shift-n (Mac) or control-shift-n (Windows)

Step 2: Create a file with the list of identifiers

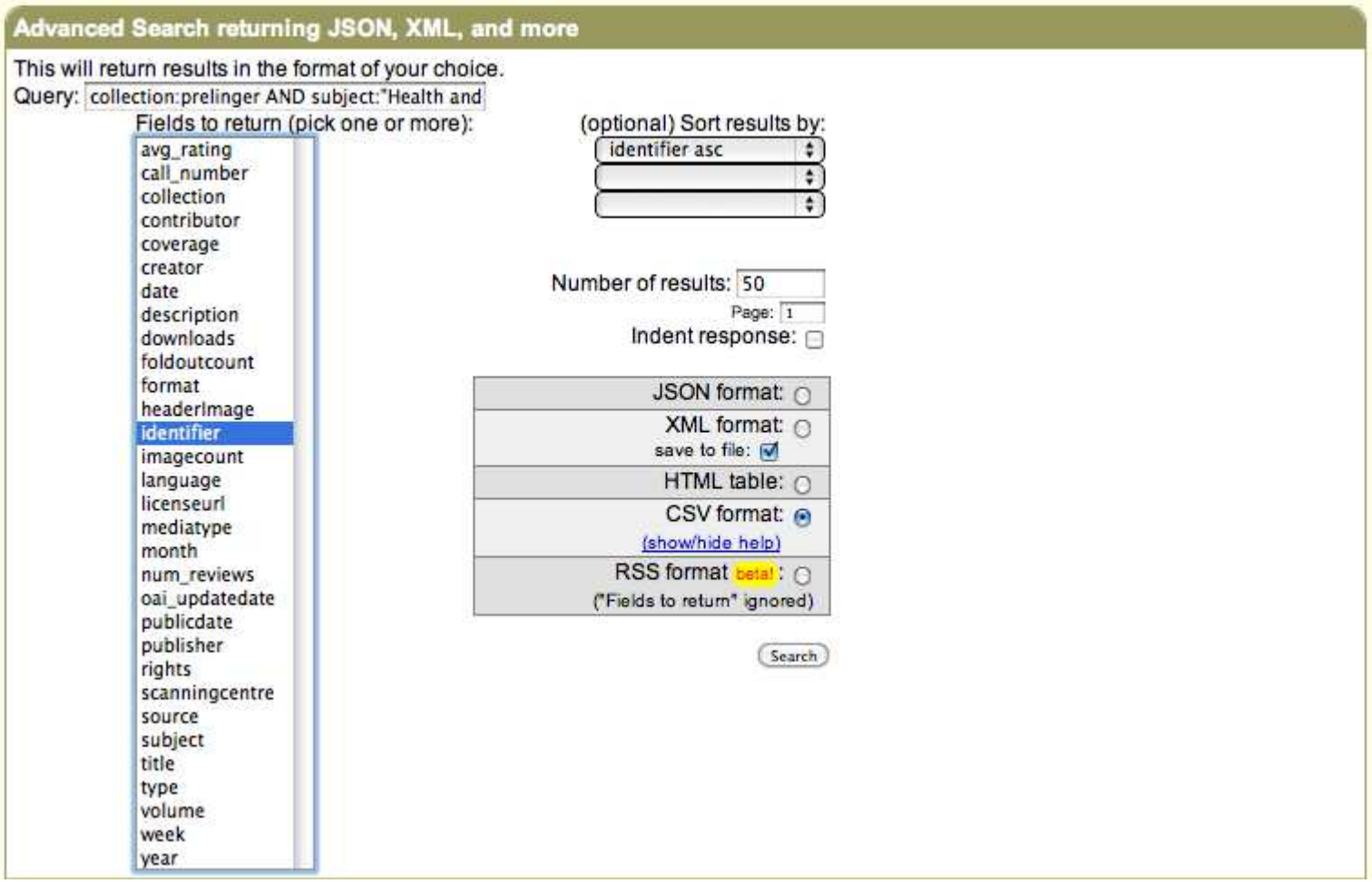
You'll need a text file with the list of archive.org item identifiers from which you want to download files. This file will be used by the wget to download the files.

If you already have a list of identifiers you can paste or type the identifiers into a file. There should be one identifier per line. The other option is to use the archive.org search engine to create a list based on a query. To do this we will use advanced search to create the list and then download the list in a file.

First, determine your search query using the search engine. In this example, I am looking for items in the Prelinger collection with the subject "Health and Hygiene." There are currently 41 items that match this query. Once you've figured out your query:

1. Go to the advanced search page on archive.org. Use the "Advanced Search returning JSON, XML, and more." section to create a query. Once you have a query that delivers the results you want click the back button to go back to the advanced search page.
3. Select "identifier" from the "Fields to return" list.
4. Optionally sort the results (sorting by "identifier asc" is handy for arranging them in alphabetical order.)
5. Enter the number of results from step 1 into the "Number of results" box that matches (or is higher than) the number of results your query returns.
6. Choose the "CSV format" radio button.

This image shows what the advance query would look like for our example:



7. Click the search button (may take a while depending on how many results you have.) An alert box will ask if you want your results – click "OK" to proceed. You'll then see a prompt to download the "search.csv" file to your computer. The downloaded file will be in your default download location (often your Desktop or your Downloads folder).

8. Rename the "search.csv" file "itemlist.txt" (no quotes.)

9. Drag or move the itemlist.txt file into your "Files" folder that you previously created

10. Open the file in a text program such as TextEdit (Mac) or Notepad (Windows). Delete the first line of copy which reads “identifier”. Be sure you deleted the entire line and that the first line is not a blank line. Now remove all the quotes by doing a search and replace replacing the ” with nothing.

The contents of the itemlist.txt file should now look like this:

```
AboutFac1941
Attitude1949
BodyCare1948
Cancer_2
Careofth1949
Careofth1951
CityWate1941
```

.....

NOTE: You can use this advanced search method to create lists of thousands of identifiers, although we don’t recommend using it to retrieve more than 10,000 or so items at once (it will time out at a certain point).

.....

Step 3: Create a wget command

The wget command uses unix terminology. Each symbol, letter or word represents different options that the wget will execute.

Below are three typical wget commands for downloading from the identifiers listed in your itemlist.txt file.

To get all files from your identifier list:

```
wget -r -H -nc -np -nH --cut-dirs=2 -e robots=off -l1 -i ./itemlist.txt -B 'http://archive.org/download/'
```

If you want to only download certain file formats (in this example pdf and epub) you should include the -A option which stands for “accept”. In this example we would download the pdf and jp2 files

```
wget -r -H -nc -np -nH --cut-dirs=2 -A .pdf,.epub -e robots=off -l1 -i ./itemlist.txt -B 'http://archive.org/download/'
```

To only download all files except specific formats (in this example tar and zip) you should include the -R option which stands for “reject”. In this example we would download all files except tar and zip files:

```
wget -r -H -nc -np -nH --cut-dirs=2 -R .tar,.zip -e robots=off -l1 -i ./itemlist.txt -B 'http://archive.org/download/'
```

If you want to modify one of these or craft a new one you may find it easier to do it in a text editing program (TextEdit or NotePad) rather than doing it in the terminal emulator.

.....

NOTE: To craft a wget command for your specific needs you might need to understand the various options. It can get complicated so try to get a thorough understanding before experimenting.You can learn more about unix commands at Basic unix commands

An explanation of each options used in our example wget command are as follows:

`-r` recursive download; required in order to move from the item identifier down into its individual files

`-H` enable spanning across hosts when doing recursive retrieving (the initial URL for the directory will be on

archive.org, and the individual file locations will be on a specific datanode)

`-nc` no clobber; if a local copy already exists of a file, don't download it again (useful if you have to restart the wget at some point, as it avoids re-downloading all the files that were already done during the first pass)

`-np` no parent; ensures that the recursion doesn't climb back **up** the directory tree to other items (by, for instance, following the "../" link in the directory listing)

`-nH` no host directories; when using `-r`, wget will create a directory tree to stick the local copies in, starting with the hostname ({datanode}.us.archive.org/), unless -nH is provided

`--cut-dirs=2` completes what -nH started by skipping the hostname; when saving files on the local disk (from a URL likehttp://{datanode}.us.archive.org/{drive}/items/{identifier}/{identifier}.pdf), skip the /{drive}/items/ portion of the URL, too, so that all {identifier} directories appear together in the current directory, instead of being buried several levels down in multiple {drive}/items/ directories

`-e robots=off` archive.org datanodes contain robots.txt files telling robotic crawlers not to traverse the directory structure; in order to recurse from the directory to the individual files, we need to tell wget to ignore the robots.txt directive

`-i ../itemlist.txt` location of input file listing all the URLs to use; "../itemlist" means the list of items should appear one level up in the directory structure, in a file called "itemlist.txt" (you can call the file anything you want, so long as you specify its actual name after -i)

`-B 'http://archive.org/download/'` base URL; gets prepended to the text read from the -i file (this is what allows us to have just the identifiers in the itemlist file, rather than the full URL on each line)

Additional options that may be needed sometimes:

`-l depth --level=depth` Specify recursion maximum depth level depth. The default maximum depth is 5. This option is helpful when you are downloading items that contain external links or URL's in either the items metadata or other text files within the item. Here's an example command to avoid downloading external links contained in an items metadata:

```
wget -r -H -nc -np -nH --cut-dirs=2 -l 1 -e robots=off -i ../itemlist.txt -B 'http://archive.org/download/'
```

`-A -R` accept-list and reject-list, either limiting the download to certain kinds of file, or excluding certain kinds of file; for instance, adding the following options to your wget command would download all files except those whose names end with _orig_jp2.tar or _jpg.pdf:

```
wget -r -H -nc -np -nH --cut-dirs=2 -R _orig_jp2.tar,_jpg.pdf -e robots=off -i ../itemlist.txt -B 'http://archive.org/download/'
```

And adding the following options would download all files containing zelazny in their names, except those ending with .ps:

```
wget -r -H -nc -np -nH --cut-dirs=2 -A "*zelazny*" -R .ps -e robots=off -i ../itemlist.txt -B 'http://archive.org/download/'
```

See http://www.gnu.org/software/wget/manual/html_node/Types-of-Files.html for a fuller explanation.

.....

Step 4: Run the command

1. Open your terminal emulator (Terminal or Cygwin)
2. In your terminal emulator window, move into your folder/directory. To do this:
For Mac: type cd Desktop/Files
For Windows type in Cygwin after the $ cd /cygdrive/c/Users/archive/Desktop/Files
3. Hit return. You have now moved into th e"Files" folder.
4. In your terminal emulator enter or paste your wget command. If you are using on of the commands on this page be sure to copy the entire command which may be on two lines. You can just cut and paste in Mac. For Cygwin, copy the command, click the Cygwin logo in the upper left corner, select Edit then select Paste.
5. Hit return to run the command.

You will see your progress on the screen. If you have sorted your itemlist.txt alphabetically, you can estimate how far through the list you are based on the screen output. Depending on how many files you are downloading and their size, it may take quite some time for this command to finish running.

.....

NOTE: We strongly recommend trying this process with just ONE identifier first as a test to make sure you download the files you want before you try to download files from many items.

.....

Tips:

- You can terminate the command by pressing "control" and "c" on your keyboard simultaneously while in the terminal window.

- If your command will take a while to complete, make sure your computer is set to never sleep and turn off automatic updates.
- If you think you missed some items (e.g. due to machines being down), you can simply rerun the command after it finishes. The "no clobber" option in the command will prevent already retrieved files from being overwritten, so only missed files will be retrieved.

This entry was posted in Technical. Bookmark the permalink.

27 Responses to *Downloading in bulk using wget*

Thomas Vanhoutte *says:*
April 29, 2012 at 10:32 pm

The power of unix, once again proven.
Thanks for the step by step explaination.

Reply

John Hauser *says:*
May 2, 2012 at 5:48 am

Thanks for this clear and detailed post!

In my case, using ubuntu 10.04, i had to add a couple of extra parameters:
-D archive.org –exclude-domains blog.archive.org –exclude-domains web.archive.org

to keep wget from wandering off into related domains like openlibrary.org, nasaimages.org and archive-it.org referred to on the http://www.archive.org home page (announcements section and /projects). The exclude-domain parameters were necessary to keep wget from recursing down into the indicated sites.

i don't know if this is a DNS name resolution artifact of running the command from outside of archive.org vs. inside, but a little experimentation with your command example got me to the answer within a couple of tries.

Reply

Jim Walton *says:*

May 11, 2012 at 4:43 am

Not getting it to work. First, the argument -cut-dirs should be –cut-dirs, at least in Windows otherwise I get an invalid argument error. When I run it I get a list of invalid URL and unsupported scheme errors. I have added the argument -A pdf to only download pdf files.

What I have done is created a csv file from advanced search and ran a script to strip off the quotes. When I run wget, it produces the url 'http://www.archive.org/download/[filename from items.txt].

Where do I go from here?

Reply

Jim Walton *says:*

May 11, 2012 at 6:20 am

–cut-dirs should read – -cut-dirs. The two dashes were converted to an em dash…

Reply

Jim Walton *says:*
May 11, 2012 at 11:09 pm

This doesn't seem to work in Windows, but works fine in Linux. Just be aware that the –cut-dirs uses 2 dashes, not a single one as it appears on the web site. The browser converts the double dash into an emdash.

In addition to the -D and –exclude-domains arguments I also added -nd so I would get all the files in a single directory instead of creating a separate directory for each file.

For those who don't normally use Linux or wget, typing "wget -help >wget.txt" without the quotes will save all the help commands in a text file in the current directory.

Reply

silvermane *says:*
May 30, 2012 at 7:25 pm

This does work on Windows, you just have to replace single with double quotes after the -B argument, and watch out for the em-dashes.

Command line examples should be placed within a [code] tag to prevent their "beautifying."

Reply

Declan Fleming *says:*

June 30, 2012 at 5:25 pm

I'm not saying this is the prettiest hack, but it finally worked on Ubuntu:

wget -r -l 1 -nc -H -np -nH -e robots=off –cut-dirs=1 -i ../itemlist.txt -B 'http://www.archive.org/download/'

the -np option was not working for me, and it kept traversing up. I locked the recursion down to 1 level and that seemed to work!

D

Reply

kumar *says:*

January 17, 2013 at 9:50 am

Hi,

I need to download a set of files which i have mentioned in a file (x.txt), after download i need all the files to be moved with today's date and i need all that files names should append to a .csv file in a single row with a space in between each file. does anyone have idea. If so help me out to fix my problem.

Reply

Jason *says:*

February 10, 2013 at 10:56 pm

You don't say what operating system you're using.
In linux something like

touch *
ls | tr “\n” “,” >text.csv
mv * [target-dir]

Reply

Pingback: *Bulk Downloading, Aaron Swartz, and Terms of Service | Internet Archive Blogs*

Pingback: *Aaron Swartz Memorial | It's My Blog, Dammit! .... Anything and Everything!*

Ralph H *says:*
February 10, 2013 at 8:40 am

This seems like an unnecessarily complex process, when all I really need is a list of archive.org URLs that I can feed to wget or another bulk HTTP downloader.
Here’s the wget output:
–2013-02-10 03:32:07– http://archive.org/details/68micro-vol-01-num-3
Connecting to archive.org|207.241.224.2|:80… connected.
HTTP request sent, awaiting response… 200 OK
Length: unspecified [text/html]
Saving to: `68micro-vol-01-num-3′
[ ] 18,499 98.3K/s in 0.2s
2013-02-10 03:32:08 (98.3 KB/s) – `68micro-vol-01-num-3′ saved [18499]
Removing 68micro-vol-01-num-3 since it should be rejected. *waste of bandwidth*

1) It appears that wget is actually downloading unrelated web pages just to retrieve the links to the PDF files. This is odd, since I specified the ‘-A’ option to only permit .pdf files – Shouldn’t I be able to search for these (or any other filetype directly?)

2) Searching for identifiers seems like the wrong thing to search for, given that I still have to allow wget to traverse a directory in hopes of finding a .pdf file.
Knowing how the web sites are structured, and the arcane list of servers and

Reply

Pingback: *Bulk Downloading Collections from Archive.org | Gareth Halfacree*

Hank Bromley *says:*
April 4, 2013 at 7:22 pm

Note that I have edited the original post, replacing “http://www.archive.org/…” with “http://archive.org/”.

Some time ago – but after I initially formulated the wget command – we arranged for http://www.archive.org requests to redirect to archive.org. I haven’t tracked down exactly why, but that redirect is apparently causing wget to see lots of additional links it wasn’t intended to, beyond those inside the targeted item directories, and consequently downloading many unrelated files. Dropping the “www.” makes it behave correctly again.

Some of the modifications suggested in previous comments were generated in order to cope with that misbehavior, and may not be necessary with the “www.” removed.

Also, the pingback just above, from Gareth Halfacree, offers a handy shell script that simplifies using the wget, as it combines the several steps into a single convenient command. Check it out!

Reply

Pingback: *450,000 Early Journal Articles Now Available | Internet Archive Blogs*

desu *says:*
April 12, 2013 at 5:35 pm

There is a wget for Windows, doesn't that make installing cygwin kinda unnecessary?

Reply

Pingback: *JSTOR "Early Journal Content" Articles Now Accessible via Internet Archive | LJ INFOdocket*

Pingback: *1923年前外文期刊:免费访问、批量下载、文本挖掘 » 编目精灵III*

Pingback: *An Aside: An Ode to Exploratory Research | Ian Milligan*

kalpaz *says:*
June 9, 2013 at 12:29 pm

i am trying to search the books which are available in pdf format, please guide me what should i put in the advance search and where

Reply

Pingback: *Finding .ca domains in the 80TB Wide Crawl | Ian Milligan*

brewster *says:*
June 26, 2013 at 1:27 pm

This is a useful tutorial on bulk downloading and then how to make a simple search engine out of the results: http://williamjturkel.net/2013/06/25/batch-downloading-and-building-simple-search-engines-with-command-line-tools-

in-linux/

Reply

Pingback: *How to use the Virtual Machine for Researchers | Internet Archive Blogs*

peng wu *says:*
July 31, 2013 at 8:08 am

Thanks for easy detail. This is a useful bulk downloading.

Reply

Pingback: *Search Engine on the WARC Collection: An Update on my Project | Ian Milligan*

Jeff Thompson *says:*
August 9, 2013 at 11:16 pm

FWIW, I found that changing `--cut-dirs=2` to `--cut-dirs=3` downloaded just the files themselves, not creating and enclosing them in directory.

Reply

google seo check *says:*
August 23, 2013 at 8:35 pm

Examples include the ‘description’ of the
content, ‘keywords’ used and the ‘Title’ of the page.

Image ALT Tags – Alt image tags are associated with images on a website. He has to deal with traffic getting back to his office.

Reply

Internet Archive Blogs
Proudly powered by WordPress.